RADC-TR-89-259, Vol IX (of twelve)
Interim Report
October 1989

NORTHEAST ARTIFICIAL INTELLIGENCE CONSORTIUM ANNUAL REPORT - 1988 Parallel Vision

Syracuse University

Christopher M. Brown and Randal C. Nelson

APPROVED FOR PUBLIC RELESE; DISTRIBUTION UNLIMITED.

This effort was funded partially by the Laboratory Director's fund.

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, NY 13441-5700

This report has been reviewed by the RADC Public Affairs Division (PA) and is releasable to the National Technical Information Services (NTIS) At NTIS it will be releasable to the general public, including foreign nations.

RADC-TR-89-259, Vol IX (of twelve) has been reviewed and is approved for publication.

APPROVED:

LEE A. RUIU
Project Engineer

APPROVED:

WALTER J. SENUS
Technical Director
Directorate of Intelligence & Reconnaissance

FOR THE COMMANDER:

IGOR G. PLONISCH
Directorate of Plans & Programs

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (IRRE) Griffiss AFB NY 13441-5700. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED

1b. RESTRICTIVE MARKINGS N/A

2a. SECURITY CLASSIFICATION AUTHORITY N/A

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE N/A

3. DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S) N/A

5. MONITORING ORGANIZATION REPORT NUMBER(S) RADC-TR-89-259, Vol IX (of twelve)

6a. NAME OF PERFORMING ORGANIZATION Northeast Artificial Intelligence Consortium (NAIC)

6b. OFFICE SYMBOL (If applicable)

7a. NAME OF MONITORING ORGANIZATION Rome Air Development Center (COES)

6c. ADDRESS (City, State, and ZIP Code) Science & Technology Center, Rm 2-296, 111 College Place, Syracuse University, Syracuse NY 13244-4100

7b. ADDRESS (City, State, and ZIP Code) Griffiss AFB NY 13441-5700

8a. NAME OF FUNDING / SPONSORING ORGANIZATION Rome Air Development Center

8b. OFFICE SYMBOL (If applicable) COES

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F30602-85-C-0008

8c. ADDRESS (City, State, and ZIP Code) Griffiss AFB NY 13441-5700

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO
62702F	5581	27	13

11. TITLE (Include Security Classification) NORTHEAST ARTIFICIAL INTELLIGENCE CONSORTIUM ANNUAL REPORT - 1988 Parallel Vision

12. PERSONAL AUTHOR(S) Christopher M. Brown, Randal C. Nelson

13a. TYPE OF REPORT Interim

13b. TIME COVERED FROM Jan 88 TO Dec 88

14. DATE OF REPORT (Year, Month, Day) October 1989 -

15. PAGE COUNT 36

16. SUPPLEMENTARY NOTATION This effort was funded partially by the Laboratory Directors' Fund. This effort was performed as a subcontract by the University of Rochester to Syracuse University, Office of Sponsored Programs.

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
12	05	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number) Artificial Intelligence, Computer Vision, Parallel Computation, Multiprocessing, Edge Detection

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The Northeast Artificial Intelligence Consortium (NAIC) was created by the Air Force Systems Command, Rome Air Development Center, and the Office of Scientific Research. Its purpose is to conduct pertinent research in artificial intelligence and to perform activities ancillary to this research. This report describes progress that has been made in the fourth year of the existence of the NAIC on the technical research tasks undertaken at the member universities. The topics covered in general are: versatile expert system for equipment maintenance, distributed AI for communications system control, automatic photointerpretation, time-oriented problem solving, speech understanding systems, knowledge base maintenance, hardware architectures for very large systems, knowledge-based reasoning and planning, and a knowledge acquisition, assistance, and explanation system.

The specific topic for this volume addresses various aspects of parallel, structural and optimal techniques in computer vision.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL Lee A. Ruiu

22b. TELEPHONE (Include Area Code) (315) 330- 4863

22c. OFFICE SYMBOL RADC (IRRE)

DD Form 1473, JUN 86 — Previous editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

Item 10. SOURCE OF FUNDING NUMBERS (Continued)

Program Element Number	Project Number	Task Number	Work Unit Number
62702F	5581	27	23
61102F	2304	J5	01
61102F	2304	J5	15
33126F	2155	02	10
61101F	LDFP	27	01

Northeast Artificial Intelligence Consortium

1988 Annual Report

Volume 9

Parallel Vision

Christopher M. Brown
Randal C. Nelson

University of Rochester
Computer Science Department
Rochester, New York 14627

This work was supported by the Air Force Systems Command, Rome Air Development Center, Griffiss Air Force Base, New York 13441-5700, under Contract Number F30602-85-0008 which supports the Northeast Aritificial Intelligence Consortium (NAIC).

Volume 9 Parallel Vision

Report submitted by
Christopher M. Brown
Randal C. Nelson
Computer Science Department
University of Rochester
Rochester, NY 14627

Table of Contents

9.1 Executive Summary

The vision group at Rochester spent the year investigating several aspects of parallel and real-time computer vision with the overall goal of implementing a set of basic sensory-motor behaviors which could serve as a foundation for more sophisticated abilities, and integrating these primary behaviors into multi-modal systems. The emphasis was on behaviors which had relevance to, and could be implemented to work robustly in, a broad range of real-world environments since these are most likely to be useful as fundamental skills.

This work reflects the position that the way to make progress in computer vision is to investigate the sensory-motor coupling that is necessary to carry out specific tasks. Once a basic behavioral repertoire is obtained, its components can be combined and modified to produce systems of increasing sophistication. This approach depends critically on the identification of appropriate foundational abilities. Since organisms, including humans, presumably evolved their present visual sophistication through a process akin to the proposed approach, one obvious source of inspiration is in the primitive visual behaviors of animals. We have concentrated on two such areas: gaze control and visual navigation.

This year's active vision work centered around commissioning the Rochester Robot, which consists of a 3 degree of freedom, two-eyed robot head connected to a Puma 761 arm. The camera-robot system is connected to a Datacube image processor, among other computational hardware, which is fast enough to allow real-time visual motion control. The robot has proved to be fruitful testbed, and a number of behaviors have been implemented, including a vestibulo-ocular reflex (VOR), vergence, and target tracking.

Research also continued into various theoretical aspects of computer vision including parallel evidence combination, parallel object recognition, principal view analysis, and adaptive Kalman filtering.

Individual activities were, briefly, as follows. Paul Chou completed his PhD and is now at IBM T. J. Watson Research Center. Paul Cooper continued his work on parallel object recognition and will finish his thesis in 1989. The Rochester Robot was commissioned with much help from Dave Tilley and Tim Becker. Ray Rimey and Rob Potter with help from Tilley and Tom Olson implemented several gaze-control mechanisms on the robot. Randal Nelson joined the faculty and the vision group this fall, coming from the University of Maryland where he worked on problems in visual navigation. He expects to continue with related work here. Chris Brown's work in 1988 divides into three phases that correspond with the calendar. From January through June he concentrated on reconstruction and segmentation algorithms implemented with Markov Random Fields and data fusion, and to a smaller extent on principle view (or aspect graph) calculations for non-convex polyhedral scenes; this work culminated in papers and the PhD thesis by Paul Chou and the paper by Nancy Watts. From June through August he concentrated on robotics and real-time vision, especially commissioning the Rochester Robot and implementing real-time vision demonstrations; this work is reported in at least three technical reports, one of which has been submitted for publication. From September through December he was at the University of Oxford, where he implemented several versions of the Kalman filter for tracking and parameter estimation, produced a technical report on the subject, implemented a simulator for the kinematic and (to some degree) control behavior of the Rochester Robot, and has been learning about adaptive and optimal control theory.

9.2 Reconstruction and Segmentation in Parallel -- Data Fusion

Integrating disparate sources of information has been recognized as one of the keys to the success of general purpose vision systems. In the work of P. Chou and C. Brown [Chou87, Chou88a, Chou88b], data fusion is used to accomplish reliable segmentation and reconstruction in parallel. The computation is formulated as a labeling problem. Local visual observations for each image entity are reported as label likelihoods. They are combined consistently and coherently in hierarchically structured label trees with a new, computationally simple procedure. The pooled label likelihoods are fused with the a priori spatial knowledge encoded as Markov Random Fields (MRFs). The a posteriori distribution of the labelings are thus derived in a Bayesian formalism. A new inference method, called Highest Confidence First (HCF) estimation, is used to infer a unique labeling from the a posteriori distribution. HCF has the computational advantages of efficiency and predictable running time. It degrades gracefully, and follows a least-commitment strategy. ITs results are consistent with observable evidence and a priori knowledge, and (not least) it out-performs other known methods. The comparative performance of HCF and other methods has been empirically tested on synthetic and real scenes, using both intensity and sparse depth data for sensor fusion experiments.

This work addressed four fundamental open questions, and provided an answer for each.

1. There was no formulation of a consistent and coherent scheme of integrating early visual observations. The contribution here was a label tree that captures increasing levels of abstraction, and a computationally simple procedure for keeping it updated as information arrives. Rather than update a "marginal belief" as evidence accumulates, which requires propagation of local belief to other sites, we maintain joint probability distributions at the sites by decoupling the notions of external evidence and prior knowledge.

2. This work successfully incorporates a priori spatial knowledge, encoded as potential energy functions that determine the MRF distribution. The results of the algorithm, i.e. the a posteriori distributions, are derived by combining the pooled external observations and the a priori information. MRFs are useful for image modeling because the prior knowledge about spatial dependencies among the entities can often be adequately modeled by considering neighborhoods that are small. Image entities like pixels are regularly structured and isotropic, which makes the priori distributions easy to formulate. The results are much more resistant to modeling error than are other methods.

3. Many algorithms for minimizing energy have been used to assign labels via MRFs: all are either slow or unreliable. The contribution of this work was the HCF algorithm, which is a quasi-deterministic steepest-descent search in an augmented label space, which includes the label "uncommitted". The HCF algorithm assigns a label to that image entity which will, when labeled, decrease the global energy the most. Thus the algorithm can be implemented with a priority queue data structure. While this is efficient on a sequential computer, techniques for parallel implementations are of interest.

4. Reconstruction algorithms (e.g. "shape from X") presume the job of segmentation is completed, and segmentation algorithms would like to use information like that provided by reconstruction. This work unifies the reconstruction and segmentation processes, using as an experimental data set intensity and sparse depth. This work requires extending HCF to deal with discrete and continuous labels (i.e. edge and depth information), and requires the use of coupled MRFs.

Results from this work are gratifying and have been reported often throughout 1988 in various locations. Included here are results for real scenes for finding boundaries (fig. 9.2.1) and for information fusion (fig. 9.2.2).

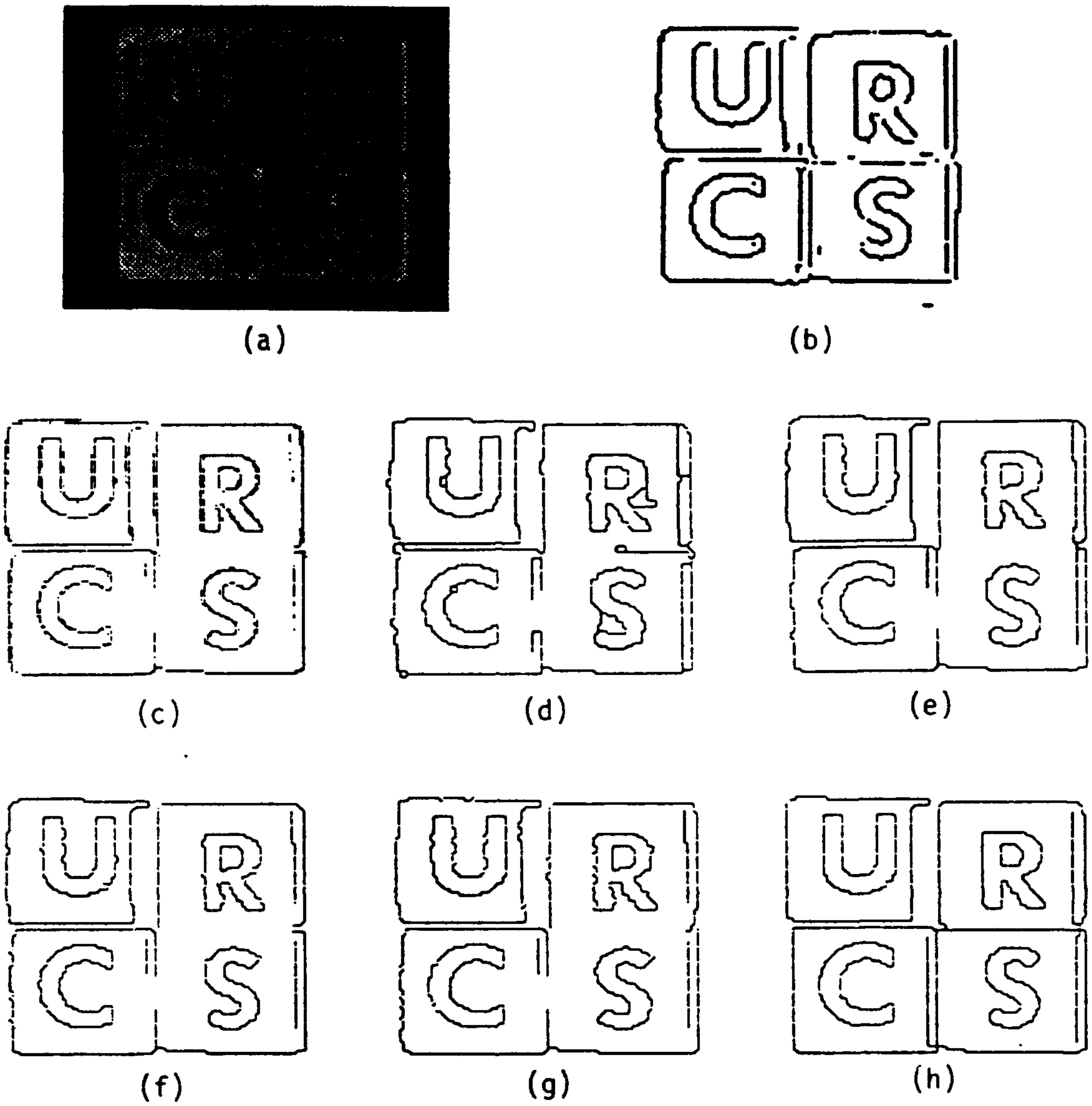


Figure 9.2.1: Boundary Detection Experiment
Data, boundary results, and resulting energy measures for labeling (the lower the better). (a) Natural 100x124 image of four plastic blocks. (b) Thinned and thresholded output of Kirsch operators. (c) TLR configuration: energy 4785 (d) Stochastic MAP estimate: energy -349. (e) Stochastic MPM estimate: energy -503. (f) ICM (scan-line visiting order) estimate: energy -503. (g) ICM (random visiting order) estimate: energy -513. (h) HCF result: energy -629.

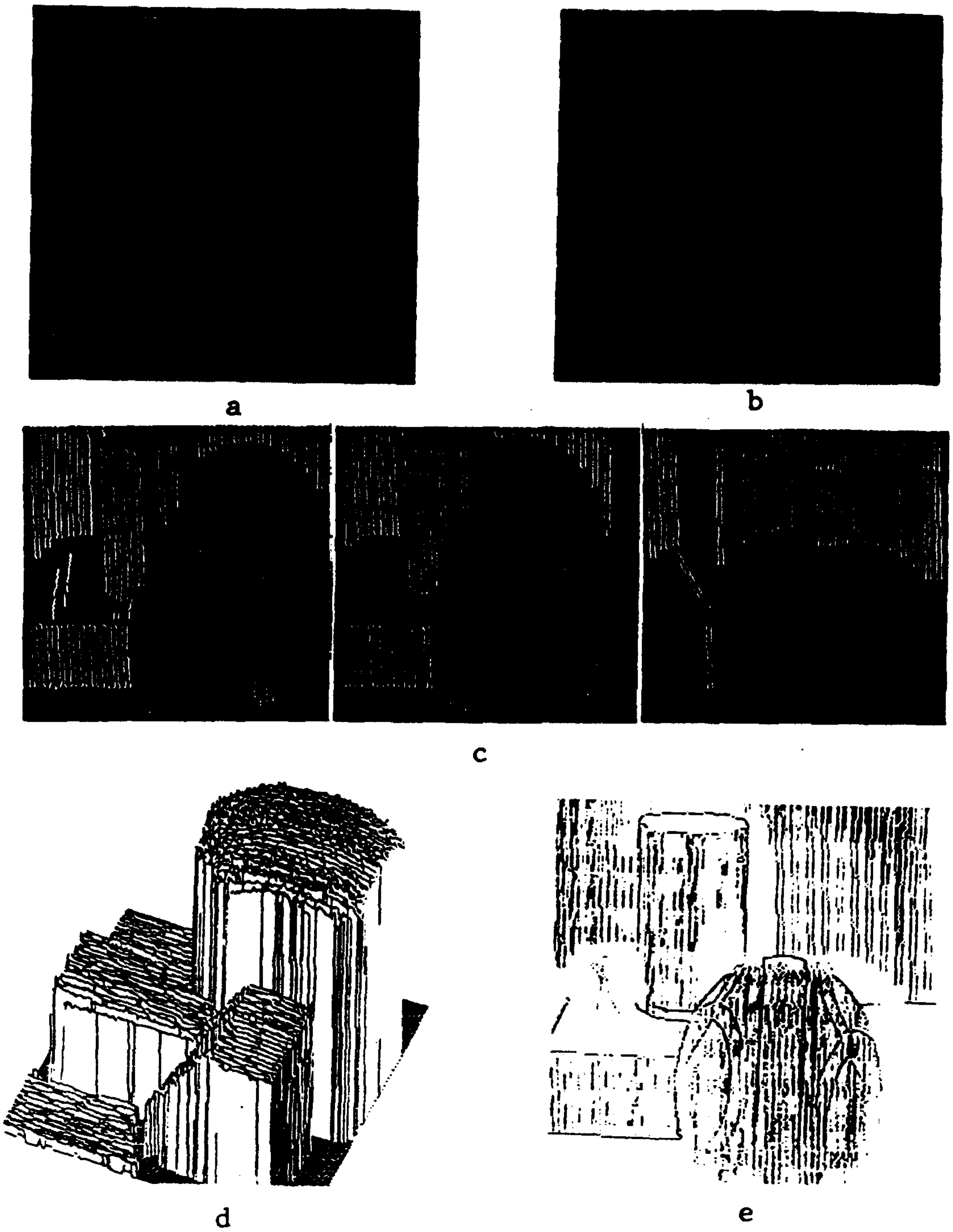
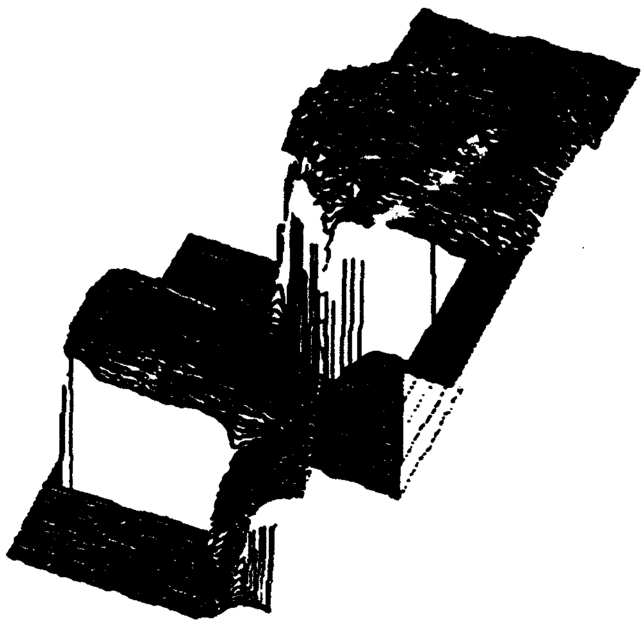
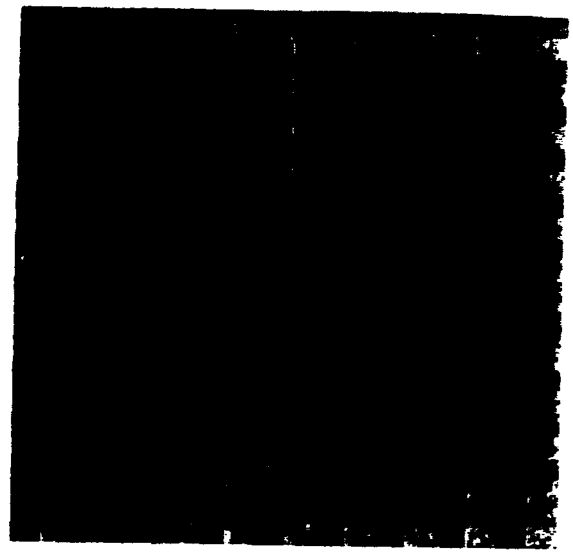


Figure 9.2.2: Experiments with Stereo Disparity Data

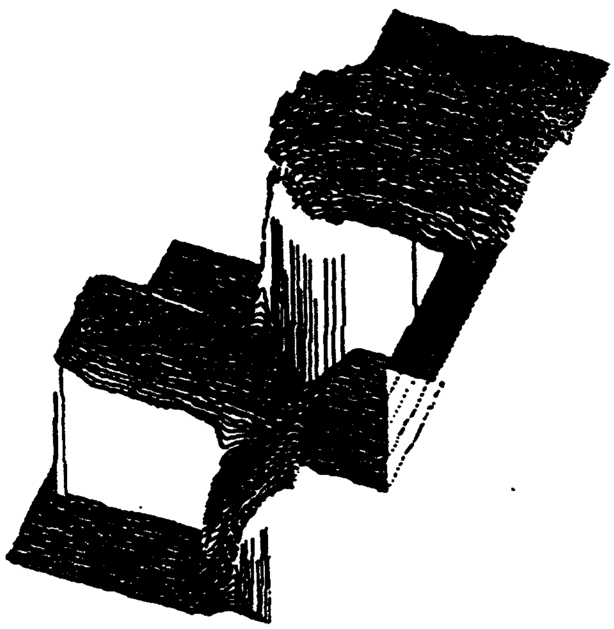
a) 200 by 200 disparity image. b) Scene with projected structured light. c) Three disparity images. d) Perspective view of the combined disparity image. e) Locations of the disparity measurements overlaid with the TLR estimate of the intensity discontinuities.



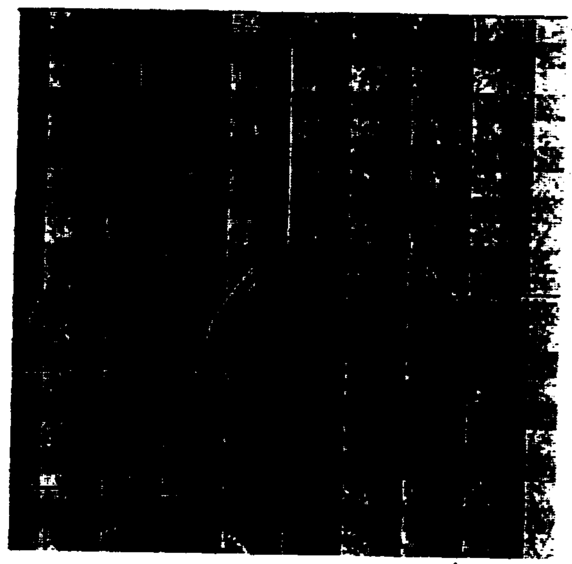
f



g



h



i

Figure 9.2.2 (Continued)
f) Reconstructed disparity map with p=0.95. g) Disparity discontinuities. h) Reconstructed disparity map with p=0.9. i) Disparity discontinuities.

9.3 Principal Views

During 1988, Nancy Watts pursued research leading to progress in the difficult problem of characterizing the different views presented to an observer (in either perspective or orthographic projection) by a non- convex polyhedron. This work was a continuation of her earlier work which produced an algorithm for computing all views of a convex polyhedron. This research is still in progress, but has resulted in a paper presented at ICPR [Watt88].

The usual approach to this problem is from the point of view of abstract computational geometry, in which existence proofs and non- constructive techniques based on them abound. Watts' work is distinguished by her desire to specify data structures and algorithms that will not only enumerate views but will allow them to be used in applications. Her earlier work interfaced nicely with a graphics program that produced sample images from any given view region for convex polyhedra.

To stand a chance of success in the violently combinatorial and geometrically complex situation that arises with non-convex objects, Watts restricted her work to a large class of objects that includes many every-day manufactured objects. She was able to catalog the incidence phenomena that take place in the projective process, and use this information to design data structures and algorithms for characterizing the aspect graph of objects in her class. The main computational tool is "plane sweeping", which is a way to keep track of regions of 3-space as their vertices are encountered by a plane sweeping through space.

Both the principal view work and the MRF work are of interest to us for their application to matching structures. Previous work by Paul Cooper, supported by the NAIC, dealt with structure matching as a recognition strategy. The work of Watts in the polyhedral domain opens the way to making complete catalogs of opaque object structure for use by matchers such as that of Cooper. Matching with a particular view gives information about the orientation, or pose, of the object matched. The MRF work of Chou can lead to probabilistic and parallel strategies for relational structure matching if the MRFs are structured non-isotropically. Thus the MRF and principal view technologies can be brought together for the eternal vision goals of object recognition and pose detection.

9.4 Parallel Object Recognition

During 1988 Paul Cooper focussed on his thesis topic which involved the general problem of parallel object recognition. The particular instance chosen for implementation was the recognition of Tinker Toy objects from images.

One development was a solution to the Tinker Toy matching problem that accommodates the geometric parameters of the object. That is, an object is recognized not just from its topology, but also from the geometric characteristics such as the lengths of the pieces and the angles between the pieces at the junction. The key to this solution was framing the labeling problem so that the geometry of the junctions was implicitly encoded. When framed in this manner, the labeling problem can be solved by the application of the massively parallel constraint satisfaction network developed earlier by Swain and Cooper [Swai88]. The application of the network to the Tinker Toy matching problem with geometry is reported in [Coop88b].

Another development was the use of domain specific information to generate optimized constraint satisfaction nets. Implementing the general form of Swain and Cooper's [Swai88] network to solve Tinker Toy matching proved infeasible due to resource requirements. But a way of exploiting the characteristics of the Tinker Toy matching domain in order to optimize the general network was developed, resulting in a smaller network that could be (and was) built. Later, a

way of specifying domain characteristics for arbitrary domains, was discovered, allowing optimized networks to be built, in an analogous manner, for any domain. This work is described in two papers by Cooper and Swain [Coop88b, Coop88c].

The final and most important development was a method for matching Tinker Toys that could incorporate inexact and uncertain information. The crux of this solution was the use of coupled Markov Random Fields to solve, simultaneously, the segmentation and matching problems in the Tinker Toy domain. The architecture of the solution is essentially the same as that of previous work [Coop88a, Coop88b], in that the problem is framed as a labeling problem in the unit/value connectionist design style. However, instead of adopting discrete constraint satisfaction [Coop88b] or discrete connectionist relaxation [Coop88a] as the formal machinery, Markov Random Fields (MRFs) are used. With a MRF representation, priors are combined with hypothesis likelihoods to yield a probability distribution of solutions with rigorous Bayesian semantics. The result is a scheme that can recognize both occluded objects, and ones obscured by noisy data.

A final report on the MRF project as well as everything else will be available in the thesis [Coop89].

9.5 The Rochester Robot

During the summer of 1988 a team at the University of Rochester commissioned the rochester robot, consisting of a Unimate PUMA 761 arm and a three-dof, two-eyed robot head [Brow88a]. The results of that effort are reported in the technical report, and have to some extent appeared elsewhere [Brow88b, Ball88, Olso88]. Thus this discussion will be brief, simply listing the accomplishments of the summer. The majority of the implementations of basic visual capabilities mirror the capabilities discussed below in section 9.7; they are basic visual reflexes or skills that we believe an active vision system can use to advantage as building blocks for behavior. In every case our goal was to produce behavior in real-time.

9.5.1 Kinematics and Coordinate Systems

Brown and Rimey undertook to decipher the documentation accompanying the robot arm and to produce a report on its semantics and syntax, and to solve some basic problems of inverse kinematics and coordinate conversion to support elementary work with the robot. This effort is documented in [Brow88b].

9.5.2 Communication between Sensing, Cognitive, Control, and Effecting Systems

Much software engineering and production effort was expended to produce basic interfaces and to program the sensori-motor interfaces. These included the MaxVideo system from Datacube, which was the province of Dave Tilley, and the communication between the host Sun computer and the Unimate, which was the province of Tim Becker.

The parallel pipelined MaxVideo system is the heart of our real-time vision capability, and it is a very complex machine with many dependencies of functionality, timing, etc. We are developing programming tools, standard configurations, and abstract models that make our life easier as users.

The robot as delivered is directly controllable only from its terminal, though communication with other computer systems is possible. Imported software from another institution, to allow control commands to be sent from the sun, proved inadequate. Tim Becker reengineered

the entire system, and his code is the basis for our current work (though it will need to be changed if we acquire a new VME-based controller).

9.5.3 Real-Time Adaptive Tracking

This capability mirrors the "smooth pursuit" system (see 9.7.). The idea here is to extract an image patch and use it as a real-time correlation template. There are, however, a number of engineering problems that give the problem a little spice. In our case there were issues of de-interlacing the video, pre-processing the input to make it more likely that the linear correlation (implemented by MaxVideo) would be adequate, and of getting the hardware to perform as it should.

Dave Tilley and Dana Ballard, with occasional kibitzing by Chris Brown and others, achieved a working system that successfully tracks moving objects. A useful capability we have not implemented is to re-acquire the correlation template in each image, thus compensating for small systematic variations of the target's appearance between frames as a consequence of its motion.

9.5.4 Vergence

This capability implements a gross vergence system (see 9.7). Here vergence is based on a global disparity calculated between subsampled left and right images. Thus it reflects large-scale image phenomena, not high-resolution ones (in this it differs from the implementation in 9.7, but the effect is the same). The work is reported in [Olso88].

The basic image-processing mechanism for implementing the global disparity calculation is the cepstral filter, which is defined as the fourier transform of the logarithm of the power spectrum. This operation is equivalent to correlating the left and right images, using a nonlinear operation to sharpen the correlation peaks. The computation leads to a measure of global disparity in image x and y, which is translated into radians of rotation via a small-angle approximation. Applying the compensating rotation verges the camera.

This system was implemented on the Euclid Digital Signal Processing computer in the MaxVideo system. It incorporates several engineering niceties, and it currently processes a pair of 32x32 subsampled images in 40 ms. There are several ideas for further work here, including using the cepstral filter in a distributed application and also producing a coarse depth map by local cepstral computations.

9.5.5 Kinetic Depth

The object of this work is to produce a depth map in real time using optic flow produced by head motions and knowledge about those head motions. The work is reported in [Ball88]. The idea is simply that the retinal flow of a patch of image of a static 3-D scene induced by a head motion depends on the depth of the scene producing the image patch and upon the head motion. It also varies with the fixation of the eyes. If the eyes fixate a patch of scene during head motion (using either tracking or vestibular feedback for example), optic flow is zero at that point. Thus with fixation kinetic depth provides depth information relative to the fixation point.

The kinetic depth calculation is a combination of simple geometry and the Horn-Shunck optic flow calculation, and results in an expression that can be implemented in lookup table. It involves derivatives of image intensity through time and intensity across space. In our hardware implementation, tracking was implemented by a loop in the MaxVideo that used hardware to report the (x,y) position of over-threshold image pixels. Derivatives were computed by

convolution on the image-processing boards, and depth was computed by a lookup table. Ideally, the system produces full-frame depth maps at pixel resolution at frame rates. Actually the amount of depth information depends on the presence of temporal and spatial derivatives in the image.

9.5.6 Vestibulo-Ocular Reflex (VOR) and Dynamic Segmentation

The VOR is a reflex that stabilizes images on the retina to compensate for head motion. Stabilization aids low-level vision by keeping edges sharp, and reducing motion blur. We noticed that motion blur could contribute positively to image segmentation if it could be used to blur objects that were NOT to be attended. Thus it would reduce high-frequency image phenomena such as edges and textures that are distracting to segmentation algorithms.

We implemented the functional equivalent of the VOR using a builtin facility of the VAL-II command language, and implemented a motion-blur amplifier in MaxVideo. The results are gratifying -- the moving head causes severe blur of scene components that are not fixated, thus throwing the fixated objects into strong relief. Of course an object may be approximately fixated before it is completely recognized or even segmented from an image, on the basis of partial information about the volume of 3-space it occupies. This work was done by Chris Brown and Ray Rimey, on the suggestion of Dana Ballard.

9.5.7 The Integration Workbench

A flexible piece of software was written by Ray Rimey and used to integrate two binocular stereo algorithms, the VOR-based segmentation tool, and to prove our understanding of the robot kinematics and coordinate systems. Other capabilities have probably been added to it by now.

The code implements the behavior of first looking around the lab for black blobs in space (these are squash balls hanging from the ceiling), and then locating them in left and right images and calculating their 3-D locations. A catalog or map of 3-D blob positions is kept, and used to perform operations on the balls such as touching them with a probe mounted to the head, or fixating them while moving and running the motion blur routine on the MaxVideo hardware, thus separating them even more from the background.

This code represents our first attempt to integrate several visual and motor capabilities in one program, and it is remarkably smooth and fast. It is the first piece of software we have written that dynamically reconfigures the connectivity (and thus function) of the MaxVideo system as a task progresses.

9.4.8 Color vision

Late in 1988 we acquired color vision capability in the form of a miniature color CCD camera which can be mounted on the robot head. One of the first applications was a real-time color histogrammer which was developed by Rob Potter. The color histogrammer runs on the MaxVideo frame-rate image processing system. It takes input from an RGB color video camera, and displays a two-dimensional projection (e.g. redness versus blueness) of the three dimensional color histogram. The histogrammer is useful for developing insights about the way the colors of a scene change as the illumination changes or as the observer moves about. The histograms themselves are expected to be useful for solving the color constancy problem and for performing fast candidate screening in multi-object recognition systems. Future research will involve training a neural net to determine and discount the illuminant in a scene from the data in the color histogram.

9.6 Kalman Filtering and Optimal Estimation Experiments

The EKF Utility software package is meant to support various aspects of current research, and thus is an evolving entity. This report is meant to give a brief snapshot of our recent work in implementing and experimenting with Kalman filter techniques. The discussion section presents opinions on the strengths and weaknesses of optimal estimation techniques in perceptual and control applications.

Kalman filtering is a form of optimal estimation characterized by recursive (i.e. incremental) evaluation, an internal model of the dynamics of the system being estimated, and a dynamic weighting of incoming evidence with ongoing expectation that produces estimates of the state of the observed system. Neither the technical details of Kalman filtering in general nor those of the variants we have employed so far will be recapitulated here. This report does provide prose descriptions, references, and examples. The primary reference is [Bar88].

9.6.1 Experiments

The basic Kalman filter is an iterative loop. Its input is the system measurements; its a priori information is the system dynamics and noise properties of system and measurement; and its useful outputs are the *innovation* (the difference between the predicted and observed measurement, by which the filter's performance may be quantified), and the estimated system state.

Our first application of the Kalman filter was inspired by example 2.4.1 in [Bar88]. Here, a target (whose state is its position and velocity) undergoes one-dimensional motion at a velocity that is constant but perturbed by white noise. Measurements yield only positional information, perturbed by an independent measurement noise process.

The (first order) Extended Kalman Filter (EKF) is a version of the Kalman filter that deals with nonlinear dynamics or nonlinear measurement equations, or both. It linearizes the problem around the predicted state (a second-order EKF makes a second-order approximation). The basic filter control loop still applies, but measurements are predicted using the nonlinear measurement equation h, and in the calculations for filter gain, state update, and covariance update, the Jacobean of h is used. Likewise state prediction is accomplished using the nonlinear state equation f and the state prediction covariance is computed using the Jacobean of f. These generalizations call for extensions to the EKF data structure in which functions (as opposed to matrices) are attached to the filter.

The sample application for the EKF was to track a moving target from a moving observer (see example 3.3.1 in [Bar88]). Again the target position evolves under noisy constant-velocity conditions. The observer travels on a course parallel to the target, but at a higher velocity, on a platform whose position is perturbed randomly by some process. The observer measures the look-down angle to the target in front, which gradually increases as the observer catches the target. The scalar angular measurement itself is noisy, and is used to estimate the target's state (position and velocity.)

The difficulty with this application is the increasing equivalent noise variance of the measurement caused by the platform perturbations, the highly leveraged positional uncertainty of the target due to noisy, low-angle measurements, and the non-linearity of the equations. Our implementation demonstrates the increasingly bad performance through time that is expected as the platform perturbations affect the measurement accuracy.

When targets *maneuver*, i.e. depart from the basic, steady-state, "normal" dynamic behavior, a tracking filter must respond. To the filter, maneuvering is signaled by a rapid increase in the normalized innovation. Recommended methods for dealing with this situation include the following.

1. Increase the process noise, or certain components of it, attributed to the target.
2. Use several filters with different assumptions in parallel, and combine their outputs probabilistically.
3. Create new filters as needed, pursuing a hypothesis tree of parallel hypotheses about target state. This tree must be pruned rapidly lest its maintenance overwhelm the computational resources.
4. Model maneuvers as colored (correlated) noise: in particular model target acceleration as a zero-mean, first-order Markov process (one with exponential autocorrelation).
5. Perform *input estimation*, in which measurements based on the nonmaneuvering model are used to detect and estimate the control input applied to the plant dynamics, and that control input in turn is used to correct the state estimate.
6. Use *variable dimension filtering* (VD filtering), in which the maneuver is considered part of the plant dynamics, not noise. Maneuver detection causes the substitution of a different, higher-order dynamic model for the lower-order, "quiescent" model.

We chose to implement the VD filter, in the light of its relatively low computational cost and relatively high efficacy (as shown by Bar-Shalom). Our illustrative application was to a target moving in two dimensions at constant velocity until some time at which it begins constant acceleration in the same direction. The quiescent filter is simply the constant-velocity target filter, the maneuvering filter is for a constant acceleration target. The example we implemented involved a target that proceeds at constant velocity for a time, then begins accelerating. The simulation demonstrates the significant improvement in tracking that accrues from switching the filter characteristics when the target maneuvers.

Tracking an object in the presence of spurious measurements (*clutter*) can be done in several ways. All assume a *validation gate* outside of which measurements are ignored: its size is a function of the desired probability of including the true measurement, and can be derived from a chi-squared calculation applied to the normalized innovation.

1. The optimal way to track a single target in clutter is the *track-splitting* approach, in which a tree of possible tracks is maintained. This can be a combinatorially expensive method.
2. An obvious alternative to treating all measurements, as it were, in parallel, is to pick a single candidate measurement and proceed as if it were the right one. The obvious candidate is the one closest in measurement space (that one with smallest normalized innovation), so this technique is called the *nearest-neighbor standard filter* (NNSF). The problem is that the true measurement can be missed.
3. A third approach is the *probabilistic data association filter* }, or PDAF. In it, the measurements within the validation gate are probabilistically blended to yield a combined innovation which is input into the Kalman filtering process. The problem is that the result does not correspond to that of any actual measurement.

General aspects of the behavior of the NNSF and PDAF filters may be predictable on abstract grounds. For instance we might make the following predictions for uniformly distributed clutter and high *probability of detection* (probability that the target is detected at all, either inside or outside the validation gate.)

1. In the presence of clutter, both the NNSF and PDAF filters are increasingly likely to "lose track" as time goes on (e.g. start tracking clutter, or have an estimate of target state outside some fixed bound).
2. With a "non-maneuvering" NNSF filter, low and high clutter levels may be less immediately harmful than medium levels, since with low clutter the target is likely to be nearest the predicted state, and with high clutter there is likely to be a clutter point near the predicted

state. It would seem that at intermediate levels the clutter would be more likely to attract the filter away from the target.

3. The NNSF filter would seem more likely to make serious errors by tracking clutter since it does not weigh the evidence. The performance of the PDAF should degrade more gracefully as conditions get worse.

We implemented the NNSF and PDAF filters, and used them to provide individual output tracks, as in the previous work. Also the programs were embedded in Monte Carlo simulations to provide data over a number of runs in statistically similar situations. The results confirm the above expectations but also provide some surprises.

We simulated individual tracking runs for situations with uniformly distributed clutter of increasing density. The actual number of clutter points in the validation gate was determined by rounding a normal variate of indicated mean (the clutter density) and standard deviation to the nearest integer, and uniformly distributing the resulting number of clutter points throughout the validation gate volume. In these runs the volume was close to unity.

In both cases, the basic filter is a constant-velocity (linear) Kalman filter. The plant noise is modeled by an acceleration component which is white noise of mean 0.0 and variance q=0.2. The clutter density standard deviation is 0.4. The validation gate size was chosen so that .99 of the target measurements should fall within it initially (as time goes on this figure seems meaningless, since the filter may quite confidently be tracking clutter). The measurement noise has variance 0.2 for the x state component and 0.1 for the y. The validation gates are rectangular (in general parallelepipedal) rather than ellipsoidal. These six figures illustrate indeed that lower clutter levels can result in worse NNSF performance than higher levels, and that performance of both filters falls off as time increases.

This work perhaps furnishes a mild surprise, *viz.* the ability of the PDAF to reacquire the track. This happens for lower clutter levels, and presumably occurs when the "signal to noise" ratio is high in the validation gate (e.g. there are few measurements in the validation gate, and at least one of them is near the actual position of the target and correctly is accorded high weight). The behavior of the PDAF in high clutter conditions is not as surprising --- it drifts, taking the average of the random clutter.

We also did statistics to quantify the number of "lost tracks", and the average final error of the estimates. The average final error function is meant to quantify the filter performance more than the discrete "lost track" measure, illustrating a linear loss function corresponding to the intuition that an estimate closer to the truth at the final timestep is better.

The results are perhaps surprising in that by this definition of lost track, the NNSF outperforms the PDAF fairly convincingly over a large range. These results are typical of many we obtained, but it is occasionally possible to engineer situations where the PDAF loses track less often. At least it seems fair to say that the situation is more complicated than it appears from Figure 6.1 of [Bar88], which indicates a marked superiority of PDAF along axes whose semantics are not clear from the text (perhaps the original paper gives more details).

The results are perhaps not surprising in that they accord with our prediction of graceful degradation of the PDAF in terms of the average error metric, using which it convincingly outperforms the NNSF. The higher sensitivity of NNSF to intermediate clutter levels is again demonstrated.

9.6.2 Conclusions

Optimal estimation techniques have at least three distinct roles to play in real-time sensorimotor systems.

1. They can be used as the basic paradigm for estimating the state of systems *internal* to the observer. Estimating external states is akin to *perception*.
2. They can be used to estimate the state of sytems *internal* to the observer. Estimating internal state involves aspects of *proprioception* (using information from internal sensors), but can also involve sensing the outside world, especially to determine dynamic observer parameters such as location and velocity.
3. They can be used as low-level utilities in service of several aspects of perception or action.

The aim of our work is to examine the characteristics of optimal estimation algorithms, including varieties of the Kalman filter, relative to the demands of a paradigm for perception and internal state estimation. The conclusion is that algorithms of the Kalman filter style are better matched to roles in internal state estimation than to paradigms of perception. Their performance as technical utilities to subserve basic sensorimotor tasks must be evaluated on a case-by-case basis.

One important role of perception is to cope with the unexpected. This seeming truism is often ignored, and has deep implications for computational perceptual models. In particular it implies that *top-down* control strategies are inadequate. Top-down (expectation-driven, hypothesis-verification) methods cope with the inherent underdetermined and computationally intensive nature of perception by using a priori knowledge to constrain the space of interpretations for perceptual data. A historic example is Shirai's polyhedral edge-following program, which "feels" its way around a polyhedral scene efficiently tracking edges it expects, but ignoring those arising from phenomena (such as holes in faces) not in its model of the polyhedral domain.

The opposite control strategy is *bottom-up*, or *data-driven*: Here the style is often a fixed order of processing of input data (say by successive levels of feature detection and extraction) leading to increasingly abstract levels of description of the input. As technology improves it is becoming possible to achieve the massive data-processing effort in real time, and the practical considerations that have partially motivated the top-down approach are vanishing (see, e.g. [Brow88a).

In one sense, the Kalman filter is an example of expectation-driven perception. In particular, the Kalman filter by definition explicitly incorporates a model of the dynamics of the plant producing the data to be interpreted. Also, faithful models of noise processes in the plant and sensor are needed if the filter is to produce optimal results. The strength of the Kalman filter for estimation is that it has these models at its disposal, but requiring them limits the sorts of perceptual jobs that the Kalman filter can reasonably be expected to perform. The problems with a strict top-down approach can be mitigated to some extent, and at some cost, by such measures as running several different filters embodying different assumptions in parallel, switching between filters when lack of fit motivates such a switch, allowing the filter to estimate control inputs to the plant, etc., as we have seen in earlier sections.

Despite such seemingly sophisticated adaptive capabilities, the extensive literature on Kalman filtering applications reveals that the perceptual tasks most often attempted are those in which the plant (often *target*) follows well-known and rather simple (e.g. ballistic) dynamic laws, and in which the target is modeled as a point in space. The typical perceptual task is tracking the target, perhaps as it maneuvers or is immersed in "clutter" (false targets). Thus the perceptual task consists of the twofold problem of linking measurements together into tracks and ignoring spurious data. In the tracking literature this is known as the *data association* problem. The basic Kalman filter mechanism provides help in the way of quantified measures of

uncertainty, surprise, information, expectation, etc. but provides nothing directly to cope with the familiar problems of controlling search in interpretation space.

Thus we see that in its role in the data association problem, which is known in computer vision as the *segmentation* problem, the Kalman filter operates in a local and bottom-up way, providing incremental datapoints to a higher-level algorithm responsible for grouping the points into coherent, semantically meaningful sets. As a result the control mechanisms in perceptual algorithms that manage the filters have a character in computer vision applications. The track-splitting, NNSF, PDAF, etc. approaches all have analogs in the edge-linking problem in computer vision, for example.

Finally, even in the constrained perceptual tasks to which they are applied, Optimal estimation schemes need a rich flow of data. Kalman filters in the literature often take on the order of tens of measurements to converge.

9.7 Robot Head and Gaze Control

Behaving, actively intelligent systems, whether mechanical or biological, must manage their computational and physical resources in appropriate ways in order to survive and to accomplish tasks. At Rochester we are building an integrated actively intelligent system that incorporates abstract reasoning (planning), sensing, and acting. The "active vision" paradigm we shall exploit incorporates the following ideas.

1. A hierarchy of control, so that the highest cognitive levels can reason in terms of "what" they want done rather than "how" to do it in detail. This hierarchy should extend throughout the system.
2. At the lower levels, the control hierarchy ends with quasi-reflexive, visual and motor capabilities or "skills". These capabilities are cooperative but to some extent independently controllable. Some are always running, and they form the building blocks on which more complex behavior is built. Examples are nulling out retinal slip to minimize motion blur, redirecting gaze as a result of attentional shifts, etc.
3. Part of the job of low-level visual capabilities is to present perceptual data, such as flow fields or depth maps, to higher-level visual processes. These processes can often benefit from knowledge of self-initiated motion on the part of the sensing entity. They can often be built on the low-level control capabilities.

The work undertaken in the latter part of 1988 was directed at the control aspects of the lower-level visual processes. In the work reported above in section 9.5, certain of these control capabilities (e.g. tracking) were implemented and used as the foundation of real-time visual skills. For instance, the ability to track was used in an algorithm that computed relative depth at frame rates [Ball88]. In the work reported below, a simulation of the robot head and eyes is used to examine the effects of different styles of interaction between certain control capabilities that are found in primates. This report simply outlines the simulation and indicates the results of the most simple possible interaction between the control loops -- no interaction at all until their outputs are summed at the effectors.

The simulation software is based on the robot kinematics derived over the summer [Brow88b], and provides a flexible tool for investigating the interaction of different control philosophies, methods, and algorithms. As of now the simulation reflects the true degree of dynamic control we can exercise over the robot: not very much. We hope, as funding permits, to implement more sophisticated controllers that give us more precise control over the robot. At that point the value of a simulator will be questionable, since it may be easier to use the device directly rather than try to simulate it at the required level of detail.

9.7.1 The Mechanism and the Imaging

The simulated mechanism is massless; this mirrors the effective behavior of our current hardware system when viewed from its high-level control operations. Its geometry captures all the essentials of the head and eye system at the U. Rochester. The robot arm is not modeled: rather the model incorporates a single head that can be positioned arbitrarily in space (six degrees of freedom: three in position, three in orientation). The model incorporates a tilt capability that affects both cameras, and a separate pan capability for each camera. The geometry of the offsets of the various axes in these links are variable, and incorporate the geometrical complexity of the real system. The independent control of the camera pans allows us to model modern theories of saccadic and vergence systems; heads with mechanical vergence capability (such as those at Harvard and Boston University) must use older models of these systems.

The system parameters assumed to controllable correspond to one set of VAL robot control parameters (corresponding to the VAL "tool coordinate system") for the head: its X,Y,Z position and A,B,C orientation. Control over the natural pans (left and right) and tilt (common) of the cameras is also assumed.

The camera models incorporate point projection with fixed focal length, as well as a "foveal-peripheral" distinction by which the location of imaged points is less certain, outside a small foveal region, depending on the off-axis angle of the target being imaged. The target itself is a single point in 3-D space, moving under dynamical laws. The experiments below were carried out with the target point in elliptical orbit about an invisible "black hole" -- thus the target followed an elliptical path.

It is assumed that the imaging system knows the distance to the target (in real life, this distance may be derived from binocular stereo, a priori knowledge, any of a number of monocular distance cues, kinetic depth calculations, etc.). It is assumed that, for each eye, the instantaneous retinal velocity of the target is known (i.e. the vector difference between its position in the current image and its position in the last image). Other than that, the system only knows the left and right image (x,y) location of the target's image. Of course the target's image position and hence image velocity is perturbed by uncertainties arising from the blurriness of peripheral vision, should the target not be foveated.

9.7.2 The Control Loops

The basic control loops that manage the system are inspired by the primate visual system. They include the following capabilities. Although not described in detail here, much is known about these control systems in animals and, to some extent, in man. Two good references are [Bert85, Pete88].

It is worth saying that the implementation of each of these control loops requires only a page of C code or so, and that at this stage most assumptions and technical decisions have been for the sake of simplicity rather than for the sake of faithfully modeling known biological systems or optimal mechanical systems. One of the major design goals, however, is that the system have enough richness to incorporate more detailed control models.

Most of the loops have several parameters, such as the proportional, integral, and derivative constants of their controllers, and their delays and latencies. Delay here means the amount of time after a commanded motion before it commences. This is often called latency in the literature. Latency refers to the time required to execute a command: it is a time constant that indicates both how soon another command can be accepted and how long the command will be affecting the controlled (velocity) variables. In the robot system the delay corresponds to how long it takes the mechanical system to respond to a motion ordered from a high software level, and the latency reflects how long it takes to complete a command. I assume that "sensors"

(actually robot and eye-control motor states read from their controllers) are available to the system immediately, without delay, and thus reflect the "true" state of the world.

There are five separate control systems.

1. Saccade:
   This is a fast slewing of cameras to point in commanded direction, during which visual processing is usually considered turned off. The command to the eyes is modeled as open loop, though there are such things as "secondary" saccades that correct errors in initial saccades. The saccadic system tries to foveate the target and to match eye rotations to the target velocity so as to be tracking the target as soon as the saccade is completed. Current opinion is that the saccadic system is aware of the 3-D location of the target, not just the location of ifs retinal image. In the implementation used for the experiments below, it operates with retinal locations and velocities, not 3-D locations or distance. The left eye is dominant in the system. The saccade aims to center the target image on the fovea of the left eye; the right eye is panned by the same amount (and of course tilted by the same amount for mechanical reasons). Thus the saccade maintains the current vergence angle. It is implemented as a constant-speed slewing of all three pan and tilt axes, with one of them attaining a system constant maximum velocity. The slewing continues until the target should be foveated (it my not be due to peripheral blurring), at which time the system is left with eye velocities that match the perceived target motion before the saccade. The saccadic system is characterized by its maximum velocity and its delay.

2. Smooth Pursuit:
   The eyes track a moving target, using retinal slip as a control input. The error here is target position in each eye, (which should be (0,0)), and the commands (here sent to both eyes independently: neither eye is dominant) are pan and tilt velocities. The pursuit system has delay, latency, and PID control.

3. Vergence:
   The vergence system measures disparity between the target position in the left and right eyes, and pans the right eye to reduce it. The vergence system has delay, latency, and PID control.

4. Vestibulo-Ocular System:
   The VOR system is open loop in the sense that its inputs come from the head positioning system and its outputs go to the eye positioning system. Its purpose is to stabilize eyes against head motion, and its inputs are thus derivatives of head position (XYZ velocities, ABC angular velocities). It also uses the distance of the target, since the appropriate response to a close target is different from that to a far target. Actually the VOR should be implemented by inverse kinematics, to which the current implementation (and presumably the neural one) is an approximation. Its output is commands to the pans and tilt controls to null out the apparent target motion caused by head motion. It is characterized by delay, latency, and open loop proportional gain.

5. Platform Compensation:
   This system is a head-control, not gaze-control system. These systems are known to interact in subtle and complex ways, but this particular reflex simply attempts to keep the eyes "centered in the head", so that the camera pans or tilts are kept within "comfortable" mechanical ranges. The "comfort function" is a nonlinear one x/((x-xmax)^2), where x is either the average pan angle or the tilt angle, and in either case xmax is the physically imposed limit of the system. This reflex is open loop (eye position affects head position), with delay, latency, and open loop proportional gain.

The system operates in two modes: smooth pursuit and saccade. During saccades, the vergence and saccade reflexes are running. While this particular disparity-driven implementation of

vergence is presumably not implemented by primates, they do have the "near triad" reflex which includes both vergence and pupil diameter reduction in saccades between near and far targets. My inclusion of the vergence during saccades is a way to implement 2/3 of the near triad. In smooth pursuit mode, the VOR, platform compensation, pursuit, and vergence systems are running.

The delays and latencies are implemented with a command pipeline, in which the commanded changes in velocities are entered opposite the time in the future they are to take effect. Time is discretized to some level. Delays result in later entry of commands. Latencies are implemented by dividing the commanded change between as many discrete time periods as necessary to spread the effect over the latency. The pipeline thus is indexed by (future) time instant, and it has entries that hold the commanded velocities for the six head degrees of freedom and three camera degrees of freedom. Each instant also has an entry corresponding to its mode (saccadic or pursuit). The pipeline has finite length, and is in fact implemented as a ring buffer.

For the experiments carried out below, the combination of control effects takes place when a reflex (say VOR) increments or decrements a velocity term in the pipeline. The increment or decrement is made to the current value which may be nonzero because of input from another reflex (say tracking). Thus the control commands are summed in the simplest possible way, as if each control system's output was a D.C. voltage and all the outputs were soldered together at the effector motor's input.

The saccadic system shuts down the pursuit system in the sense that, for the duration of the saccade (which is computed from the image distance it must move the fovea and the maximum velocity it can move), all other commands in the pipeline are overwritten, and the mode is changed to "saccade". Further commands trying to affect these instants are ignored. If the system is in pursuit mode, command velocities are summed as mentioned in the previous paragraph. The system is diagramed in the flow chart shown in figure 9.7.1.

9.7.3 An Experiment on Control Interactions

The following sequence of graphs illustrates the cumulative effect of adding control capabilities in the manner outlined above: they operate independently and their outputs are simply summed at the effectors. For this run the delays and latencies were both set to constant low values. Large delays would have a very serious effect on the performance of the system, and is a serious issue. In the work at Harvard it is addressed by positive feedback, which reduces the delay effects (and is also good for putting actions into "head frame" instead of "retinal frame").

In each case, the system is tracking (or acquiring and tracking) a target moving around in an ellipse whose plane is parallel to the image plane. Each graph plots tracking error in x and y as a function of time for the left and right eye. Since the goal is to foveate the target at (0,0), the graphs also show the image coordinates of the target through time. There are actually four graphs per figure, but often the left and right y errors are identical because both cameras have the same tilt. Basically, figures 9.7.2 - 9.7.7 illustrate the increasingly effective final behavior of the system as reflexes are added.

9.7.4 Learning and Adaptation

The simulated system can support other relevant aspects of the control problem, including the important one of adapting to changes in the plant. The one application here that I have implemented involves using "the MIT rule", which is a gradient descent method similar to back-propagation learning in neural nets, to learn part of the robot head geometry. In a way this system acts like another control system, which inputs the discrepancies between expected and

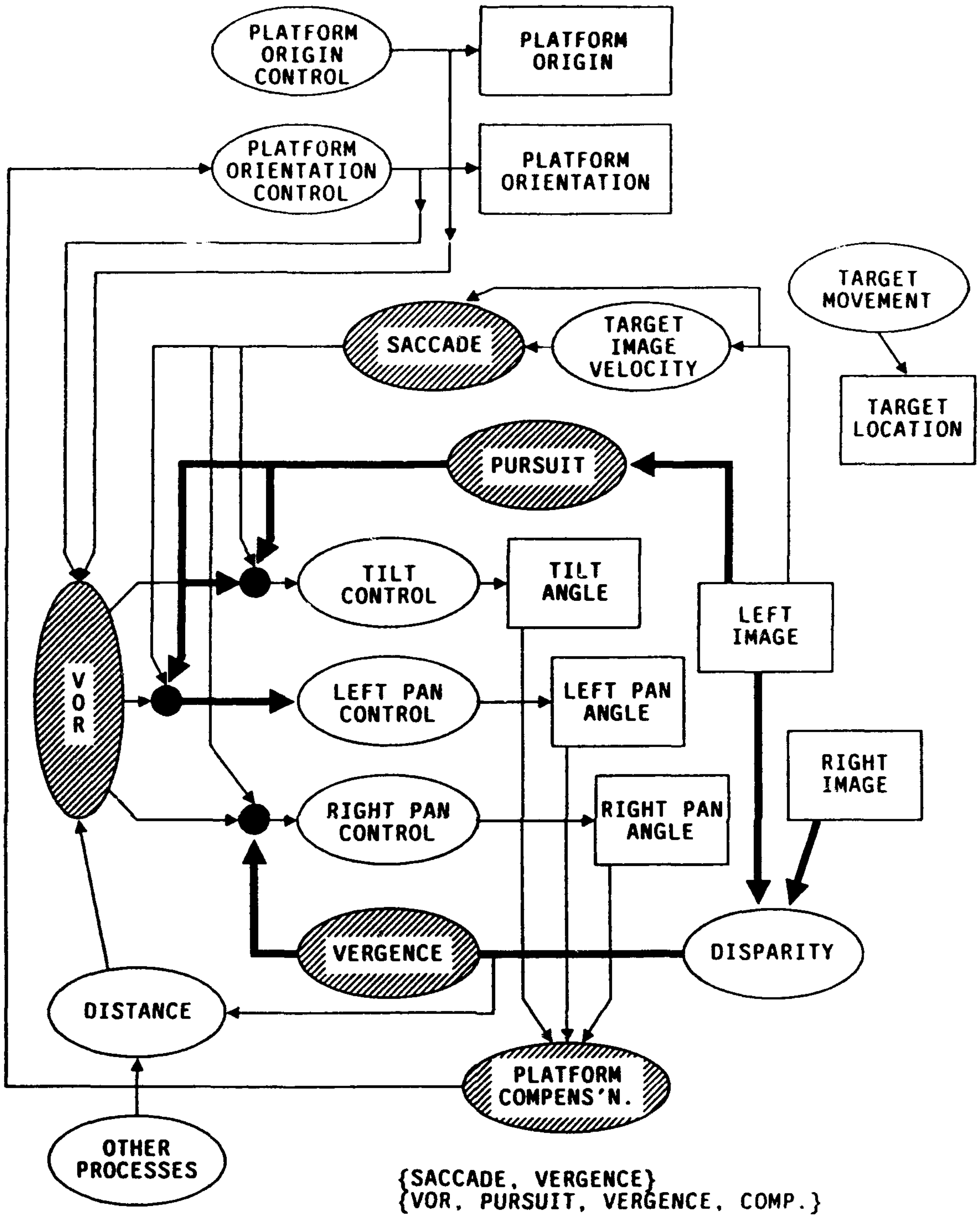


Figure 9.7.1: The five control loops in the robot head simulation.

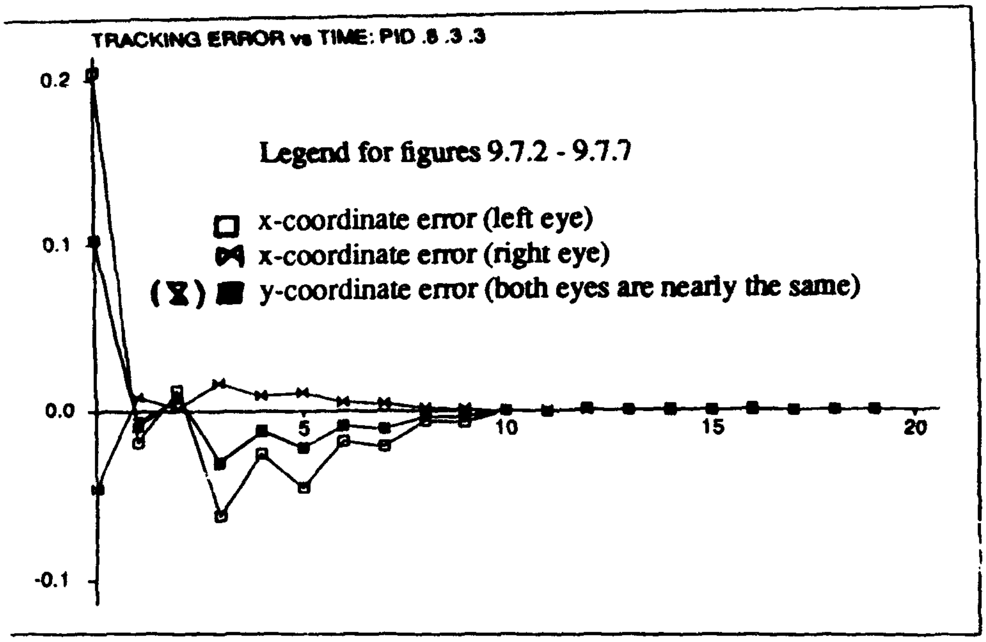


Figure 9.7.2: Tracking only
Cameras are assuming mechanically impossible pans and tilts for which I am not checking explicitly. If I were modeling maximum excursion, tracking error would at some point rapidly increase as they cameras hit their stops.

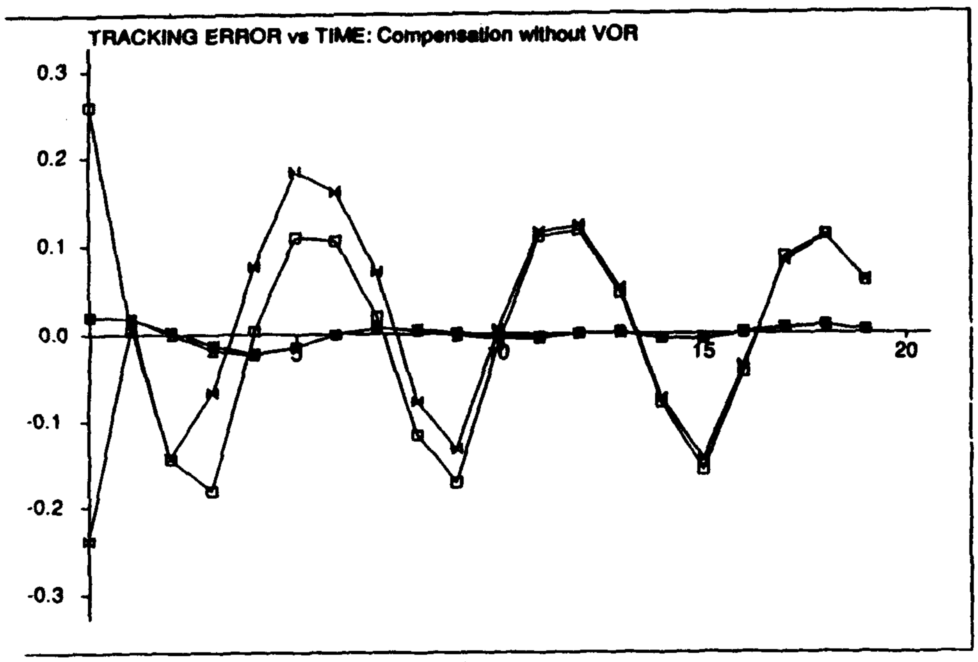


Figure 9.7.3: Head compensation added
To keep the eyes centered in the head, this reflex moves the head in the same direction the eyes are moving. Unfortunately in this case such a move amplifies the tracking corrections, overcompensates, and renders the system unstable.

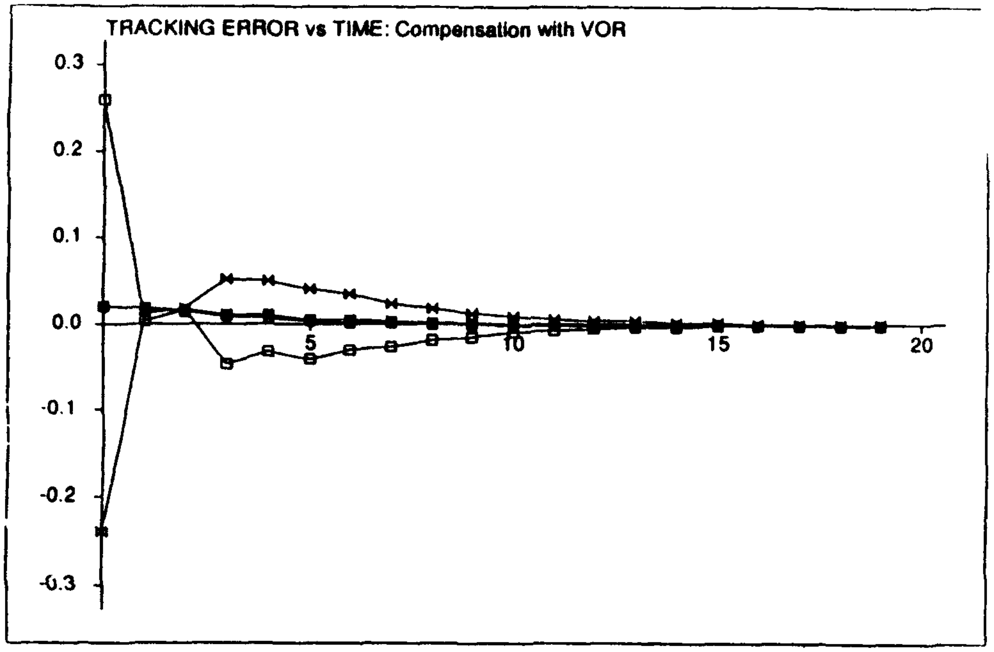


Figure 9.7.4: VOR added

VOR's function is to compensate for head motion. In this case the head motion was engendered by the head compensation reflex, which itself was driven by the tracking motions of the eyes. VOR effectively stabilizes the system, which may now be imagined as moving both its eyes and head to track the object.

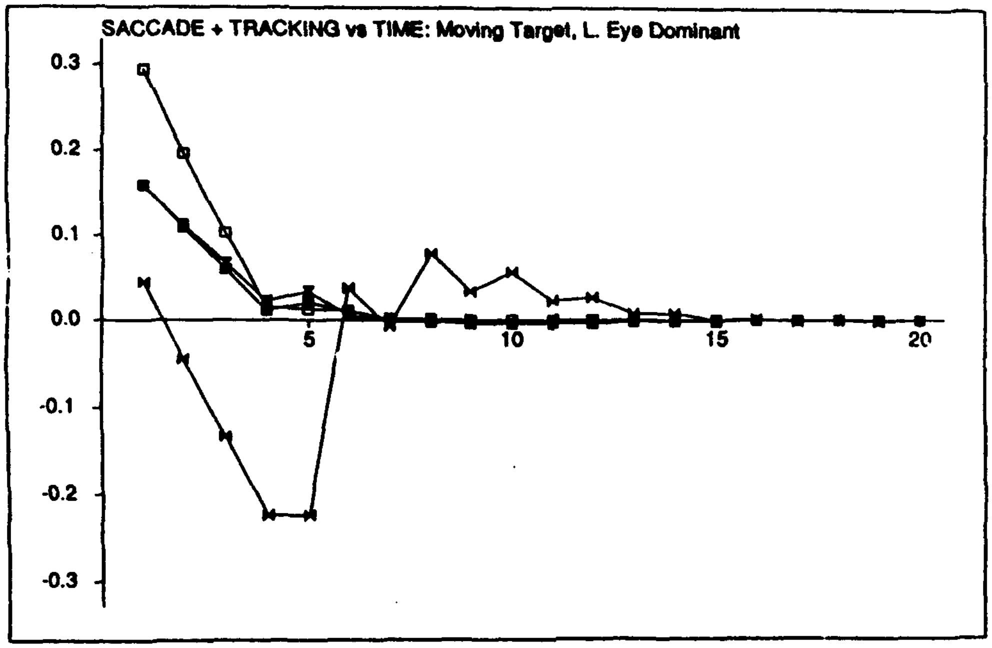


Figure 9.7.5:

Here the system makes an initial saccade to place the target on the fovea and match its velocity. The left eye dominance and the implementation that preserves vergence during saccades (the R eye gets the same pan velocity signal) means that the right eye is slewed off target as the left eye is slewed on target.

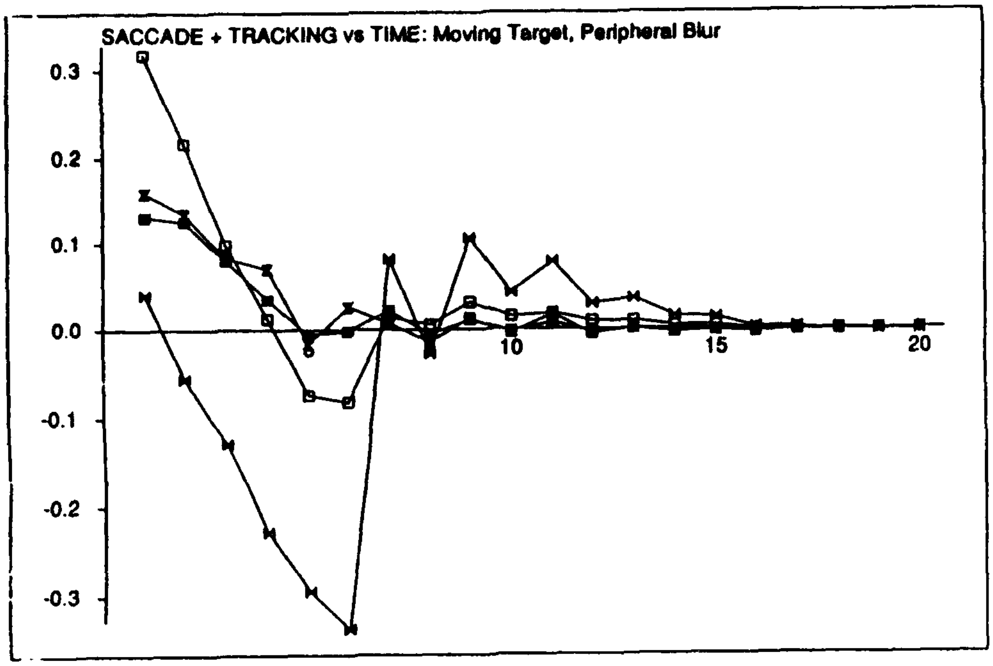


Figure 9.7.6:
For this experiment the effects of peripheral blurring were increased, so the saccade is actually made to an inaccurate location.

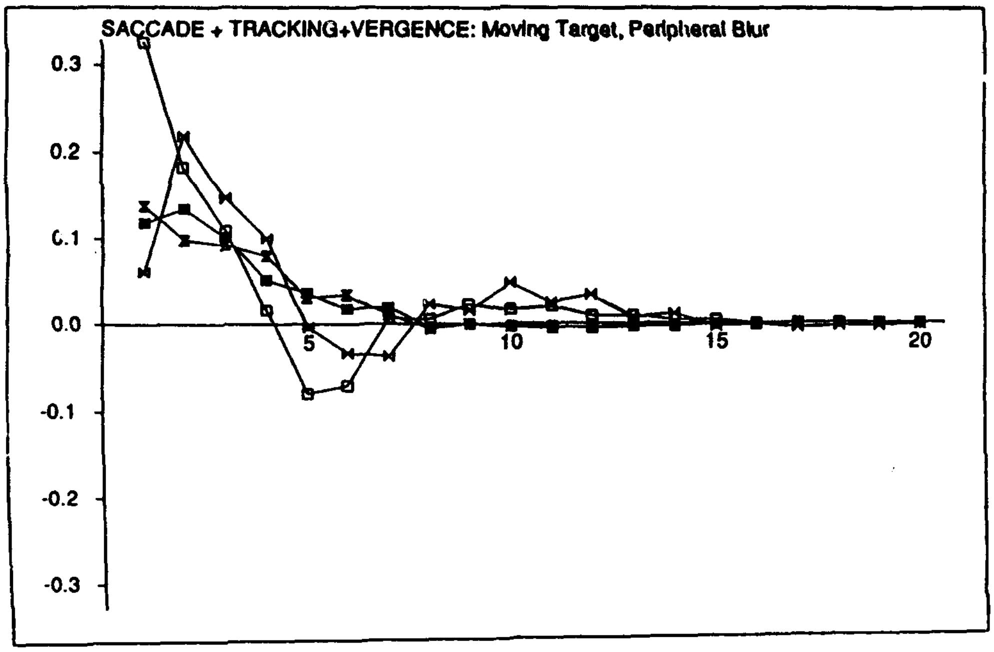


Figure 9.7.7:
Here, conditions were as for Fig. 9.5.6 but the vergence system was turned on. It tries to drive thc target image's disparity between the left and right eyes to zero, and results in improved performance.

observed target motions given eye motions, and outputs are parameters to the modeled plant (in this case, lengths of links in the head kinematic chain).

9.8 Visual Navigation

In fall 1988 Randal Nelson joined the faculty and the vision group at Rochester having completed his PhD at the University of Maryland. The dissertation research involved the description and implementation of a set of foundational abilities for visual navigation. In particular, visual methods were described for performing passive navigation, obstacle avoidance, and homing in general, real-world environments [Nels88d]. This work fits nicely within the framework of active vision which the group is currently pursuing and, since he intends to continue work along similar lines, a summary of the dissertation results follows.

Passive navigation is a process by which a system obtains information about its rotation and translational motion. This information is useful in navigation to stabilize and direct the motion of the system. Visual methods attempt to obtain the motion parameters from a time-series of images [Gibs50, Praz80, Horn81, Hild83, Lawt83, Long84, Koen86]. The problem is hard because solution methods tend to be extremely sensitive to small errors in the input, while accurate image flow or point correspondence information is difficult to obtain [Tsai81, Adiv85, Anan85, Nage86, Verr87]. The dissertation shows how accurate motion parameters can be obtained from inaccurate flow data by utilizing image information over the entire visual sphere [Nels88a]. Essentially global topological constraints are used to stabilize the process. It is interesting to note that such spherical images are available to flying insects such as bees and dragonflies, so there is a biological precedent.

Obstacle avoidance refers to the ability of a system to move about in the environment without striking the objects in it. This is a fundamental navigational behavior. It is shown that computation of divergence-like properties of the visual flow field provides qualitative cues which are invariant under rotation of the system and which are sufficient to permit the system to avoid collisions. The method is applicable in general environments, the only requirement being the presence of sufficient visual texture to allow the image flow to be roughly approximated. Empirical measurements show that sufficient texture is present in ordinary objects such as stones, trees, and faces. The method was implemented and used successfully to control the motion of a camera in various environments.

Homing is the process by which an autonomous system guides itself to a particular location on the basis of sensory input. This is a slightly more sophisticated, but still fundamental navigational ability. In the dissertation, a method of visual homing using an associative memory [Hint81, Ackl85, Rume86, Smol86,] based on a simple pattern classifier is described [Nels88c]. Homing is accomplished without the use of an explicit world model by utilizing direct associations between learned visual patterns and system motor commands. The technique is analyzed in terms of a pattern space and conditions obtained which allow the system performance to be predicted on the basis of statistical measurements on the environment. The method was implemented and used to guide a robot-mounted camera in a three-dimensional environment.

9.9 References

[Ackl85] - D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, A learning algorithm for Boltzmann machines, *Cognitive Science* 9, 1985, 147-169.

[Adiv85] - G. Adiv, Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field, Proc. 1985 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 70-77.

[Anan85] P. Anandan and R. Weiss, Introducing a smoothness constraint in a matching approach for the computation of optical flow fields, Proc. Third Workshop on Computer Vision: Representation and Control, 1985, 186-194.

[Ball88] D. H. Ballard and A. Ozcandarli, Eye fixation and early vision: Kinetic depth, Proc., 2nd IEEE Int'l. Conf. on Computer Vision, December 1988.

[Bar88] Y. Bar-Shalom and T. E. Fortman, *Tracking and Data Association*, Academic Press, 1988.

[Bert85] *Adaptive Mechanisms in Gaze Control*, Edited by Berthoz and Melvill-Jones, Elsvier 1985,

[Broo86] R. A. Brooks, Achieving artificial intelligence through building robots. A. I. Memo 899, Massachusetts Institute of Technology Artificial Intelligence Laboratory, May, 1986.

[Brow88a] C. M. Brown (Ed), with D.H. Ballard, T.G. Becker, R.F. Gans, N.G. Martin, T.J. Olson, R.D. Potter, R.D. Rimey, D.G. Tilley, and S.D. Whitehead, The Rochester obot, TR 257, Computer Science Dept., U. Rochester, August 1988.

[Brow88b] C. M. Brown and R.D. Rimey, Coordinates, conversions, and kinematics for the Rochester Robotics Lab, TR 259, Computer Science Dept., U. Rochester, August 1988.

[Coop88a] Paul R. Cooper, Structure Recognition by Connectionist Relaxation: Formal Analysis, Proceedings Canadian Artifical Intelligence Conference CSCSI-88, Edmonton, Alberta, June 1988.

[Chou87] P. B. Chou, and C.M. Brown, Probabilistic information fusion for multi-modal image segmentation, Proc., Int'l. Joint Conf. on Artificial Intelligence, Milan, August 1987.

[Chou88a] P. B. Chou and C.M. Brown, Multimodal reconstruction and segmentation with Markov Random Fields and HCF optimization, Proc., DARPA Image Understanding Workshop, Boston, MA, April 1988; submitted (August 1988), *Int'l. J. of Computer Vision.*

[Chou88b] P. B. Chou, *The theory and practice of Bayesian image labeling*, TR 258 and PhD. Thesis, Computer Science Dept., U. Rochester, August 1988; submitted, *Int'l. J. Computer Vision.*

[Coop88b] Paul R. Cooper and Michael A. Swain, Parallelism and Domain Dependence in Constraint Satisfaction, University of Rochester Technical Report TR 255, December 1988.

[Coop88c] Paul R. Cooper and Michael A. Swain, Domain Dependence in Constraint Satisfaction, submitted for publication, December 1988b.

[Coop89] Paul R. Cooper, *Parallel Object Recognition for Structure*, Ph.D. Thesis, May 1989.

Gibs[50] J. J. Gibson, *The Perception of the Visual World*, Houghton Mifflin, Boston, Mass., 1950.

[Hild83] - E. C. Hildreth, *The Measurement of Visual Motion*, The MIT Press, Cambridge, Mass., 1983.

[Hint81] - G. E. Hinton and J. A. Anderson (editors), *Parallel Models of Associative Memory* Lawrence Earlbaum Associates Inc., Hillsdale, N.J., 1981.

[Horn81] - B.K.P. Horn and B.G. Schunk, Determining optical flow, *Artificial Intelligence* **17**, 1981, 185-204.

[Koen86] J. J. Koenderink, Optic flow, *Vision Research* **26**, 1986, 161-180.

[Lawt83] - D.T. Lawton, Processing translational motion sequences, *CVGIP* **22**, 1983, 116-144.

[Long84] H. C. Longuet-Higgins K. and Prazdny, The interpretation of a moving retinal image, *Proceedings of the Royal Society of London* **B 208**, 1984, 385-397.

[Marr82] D. Marr, *Vision*, W. H. Freeman & Company, San Francisco, 1982

[Nage86] H. H. Nagel and W. Enkelmann, An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. PAMI* **85**, Sept. 1986, 565-593.

[Nels88a] R.C. Nelson and J. Aloimonos, Finding motion parameters from spherical flow fields (or the advantages of having eyes in the back of your head) *Biological Cybernetics* **58**, 1988, 261-273.

[Nels88b] R.C. Nelson and J. Aloimonos, Using flow field divergence for obstacle avoidance in visual navigation, *IEEE transactions on PAMI*, To appear.

[Nels88c] R.C. Nelson Visual homing using an associative memory, *International Journal of Computer Vision*, Under review.

[Nels88d] R.C. Nelson *Visual Navigation*, PhD dissertation, University of Maryland, 1988.

[Olso88] T. J. Olson, and R.D. Potter, Real-time vergence control, TR 264, Computer Science Dept., U. Rochester, November 1988; submitted, IEEE Conf. on Computer Vision and Pattern Recognition.

[Pete88] *Head Control*, Ed. by E. Peterson, Oxford U. Press 1988.

[Poes88] Massimo Poesio, Using Markov Random Fields for Segmentation in the Tinker Toys Domain, University of Rochester Internal Report, December 1988.

[Praz80] K. Prazdny, Egomotion and relative depth map from optical flow, *Biological Cybernetics* 36, 1980, 87-102.

[Rume86] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, Learning internal representations by error propagation, in *Parallel Distributed Processing*, D. E Rumelhart and J. L McClelland, editors, The MIT Press, Cambridge Mass., 1986, 318-362.

[Smol86] - P. Smolensky, Information processing in dynamical systems: foundations of harmony theory, in *Parallel Distributed Processing*, D. E Rumelhart and J. L McClelland, editors, The MIT Press, Cambridge Mass., 1986, 195-281.

[Swai88] Michael A. Swain and Paul R. Cooper, Parallel Hardware for Constraint Satisfaction, Proceedings AAAI-88, American Association of Artificial Intelligence, St. Paul, Minn., August 1988.

[Tsai81] R. Y. Tsai and T. S. Huang, Estimating 3-D motion parameters of a rigid planar patch I, *IEEE ASSP* 30, 1981, 525-534.

[Verr87] A. Verri and T. Poggio, Against quantitative optical flow, International Conference on Computer Vision, June 1987, 171-180.

[Watt88] N. Watts, Calculating the principal views of a polyhedron, TR 234, Computer Science Dept., U. Rochester, December 1987; Proc., Int'l. Conf. on Pattern Recognition, November 1988.

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control, Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. The areas of technical competence include communications, command and control, battle management information processing, surveillance sensors, intelligence data collection and handling, solid state sciences, electromagnetics, and propagation, and electronic reliability/maintainability and compatibility.